MOBILE APPLICATION DEVELOPMENT USING HTML, CSS, JAVASCRIPT AND PHONEGAP BUILD

Mahmud Al-Kauthar (kauthar.net)

Table of Contents

Tutorial 01 – Introduction

MOBILE APPLICATION (app) is a software application developed specifically for use on small or wireless computing devices, such as smartphones and tablets, rather than desktop or laptop computers.

Mobile app operating system (OS) such as:

- Android (Google)
- iOS (Apple)
- BlackBerry OS (BlackBerry)
- Windows Phone (Microsoft)

Mobile app types can be generally broken down into:

Android iPhone / iPad BlackBerry Windows Phone

- Native Apps
- Web Apps
- Hybrid Apps

NATIVE APPS - are built for a specific platform with the tools provided by the platform vendor. Example:

- iOS using Xcode / Objective-C, Swift
- Android using Android Studio / Java
- Windows Phone using Visual Studio / C#

Native apps take full advantage of the operating system by:

- Directly interfacing with device hardware like the camera, accelerometer, compass, or GPS
- Accessing data and information on the phone such as contacts, photos, videos, and music

WEB APPS - are application program that is stored on a server and delivered over the Internet through a browser interface.

Web apps are built with any server-side technology such as PHP, Java or ASP.NET that render HTML code to perform well on a device browser.

HYBRID APPS – are web application built with web technologies (HTML, JavaScript, CSS) that is then "wrapped" in native device code to extend the functionality and availability of the app

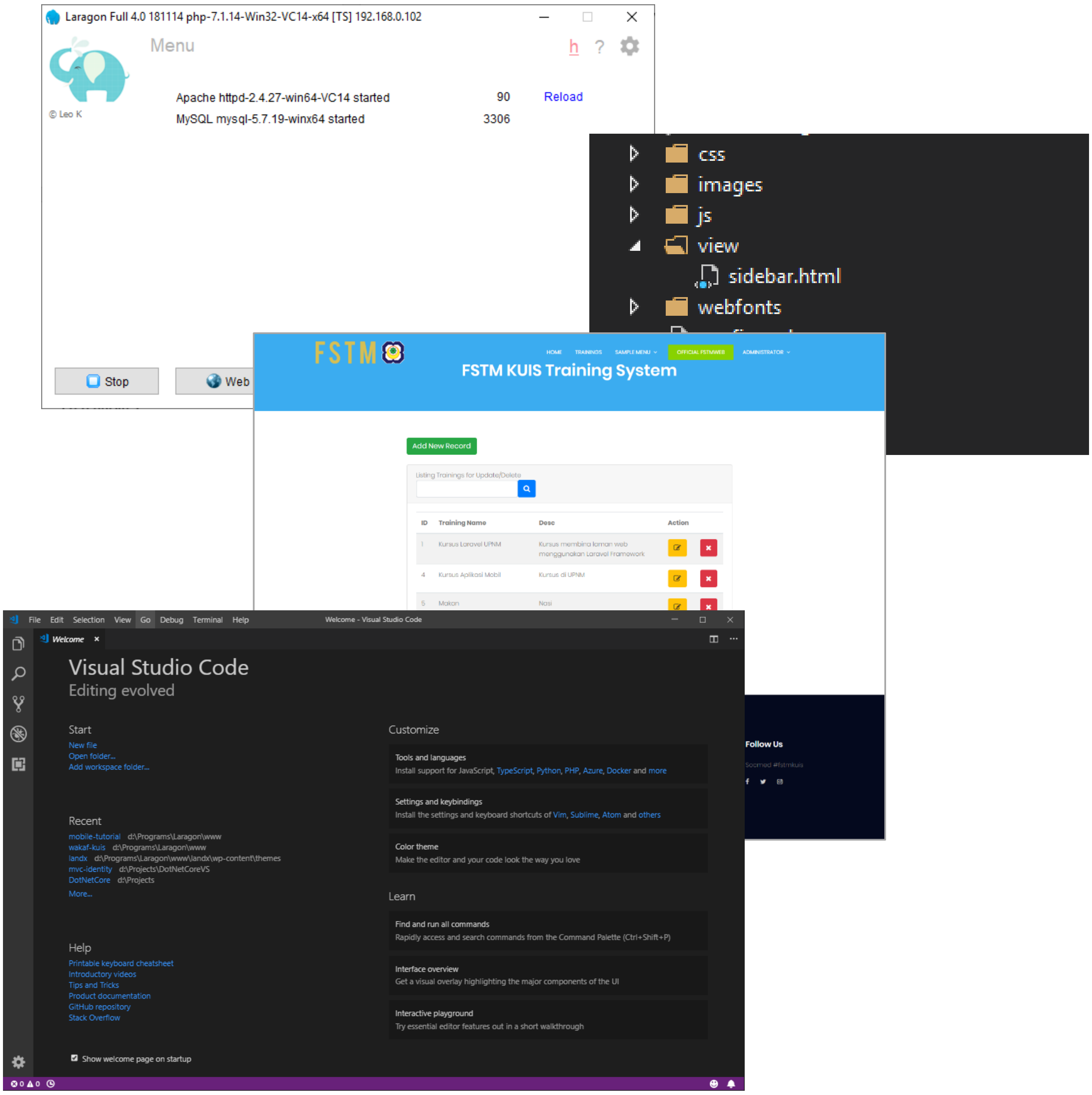
It is the marriage of web technology and native execution.

Tutorial 02 – Setup Development Environment

In this chapter, we will start by setup our development environment. For the development, we are using HTML, CSS, JavaScript, jQuery library to achieve our target.

Prerequisites

1- Laragon – Use to send and retrieve data from server using Ajax and REST API
2- Template App – Download mobile app template skeleton from the link given by the instructor
3- Editor – Visual Studio Code, Sublime, Notepad++
4- Server Side – Backend of the system which hold the data and as API provider



Laragon Full 4.0 181114 php-7.1.14-Win32-VC14-x64 [TS] 192.168.0.102

Apache httpd-2.4.27-win64-VC14 started	90	Reload
MySQL mysql-5.7.19-winx64 started	3306	

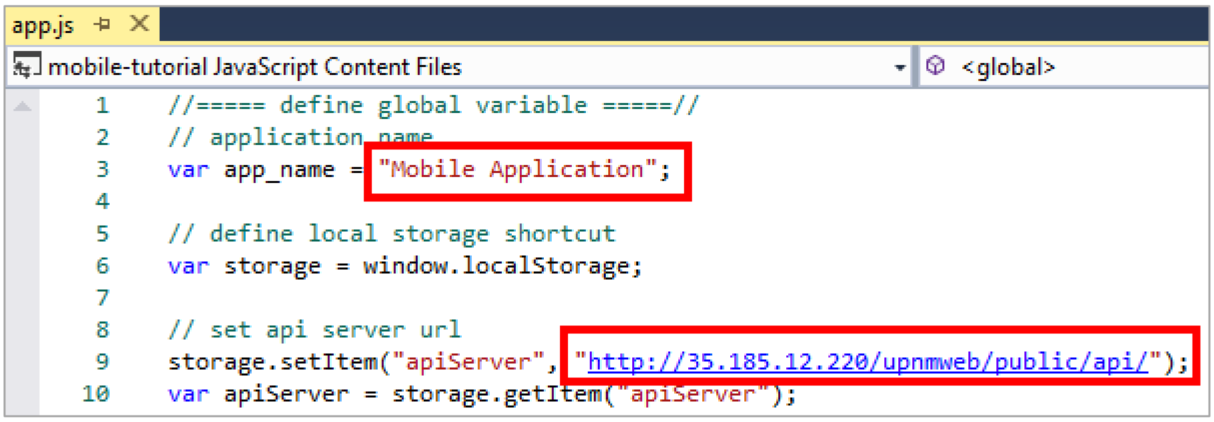
Menu
© Leo K
Stop
Web
css
images
js
view
sidebar.html
webfonts
FSTM
HOME TRAININGS SAMPLE MENU OFFICIAL FSTMWEB ADMINISTRATOR
FSTM KUIS Training System
Add New Record
Listing Trainings for Update/Delete
ID Training Name Desc Action
1 Kursus Laravel UPNM Kursus membina laman web menggunakan Laravel Framework
4 Kursus Aplikasi Mobil Kursus di UPNM
5 Makan Nasi
Follow Us
File Edit Selection View Go Debug Terminal Help
Welcome - Visual Studio Code
Welcome
Visual Studio Code
Editing evolved
Start
New file
Open folder...
Add workspace folder...
Recent
mobile-tutorial d:\Programs\Laragon\www
wakaf-kuis d:\Programs\Laragon\www
landx d:\Programs\Laragon\www\landx\wp-content\themes
mvc-identity d:\Projects\DotNetCoreVS
DotNetCore d:\Projects
More...
Help
Printable keyboard cheatsheet
Introductory videos
Tips and Tricks
Product documentation
GitHub repository
Stack Overflow
Show welcome page on startup
Customize
Tools and languages
Install support for JavaScript, TypeScript, Python, PHP, Azure, Docker and more
Settings and keybindings
Install the settings and keyboard shortcuts of Vim, Sublime, Atom and others
Color theme
Make the editor and your code look the way you love
Learn
Find and run all commands
Rapidly access and search commands from the Command Palette (Ctrl+Shift+P)
Interface overview
Get a visual overlay highlighting the major components of the UI
Interactive playground
Try essential editor features out in a short walkthrough

Application Configuration

Extract ***mobile-app-template.zip*** inside www folder in Laragon.

Open the extracted folder using editor.

Go to 'js' folder and open ***app.js*** file.

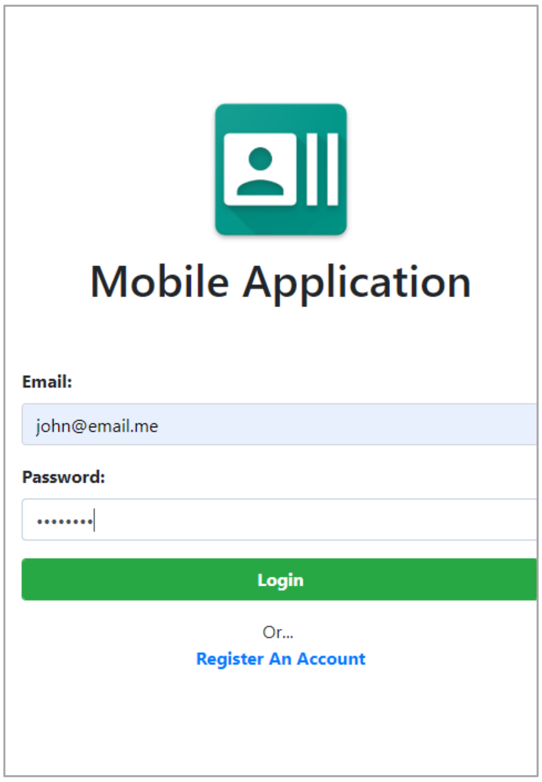


```
//===== define global variable =====//
// application name
var app_name = "Mobile Application";

// define local storage shortcut
var storage = window.localStorage;

// set api server url
storage.setItem("apiServer", "http://35.185.12.220/upnmweb/public/api/");
var apiServer = storage.getItem("apiServer");
```

Update application name and REST API destination link.

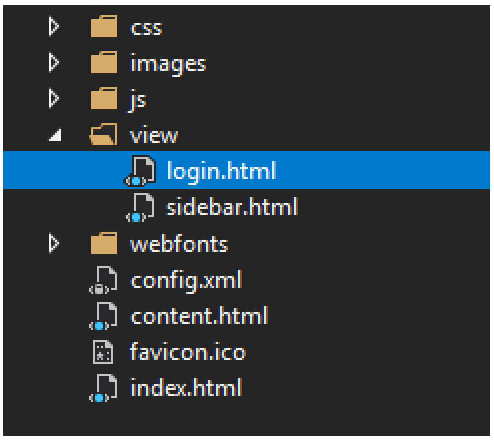
Tutorial 03 – Login / Logout Page

Figure below is the login page when user open this application.

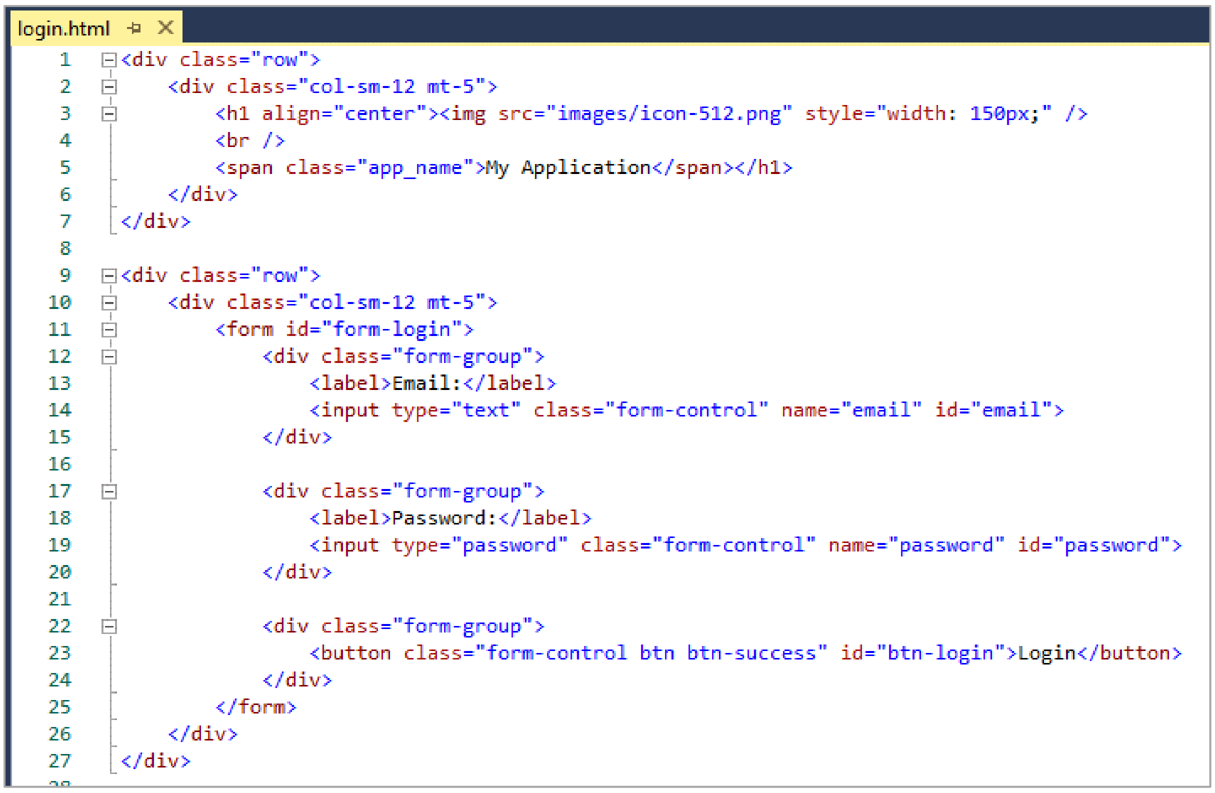


Login

Create new HTML file with name ***login.html*** inside 'view' folder as below.



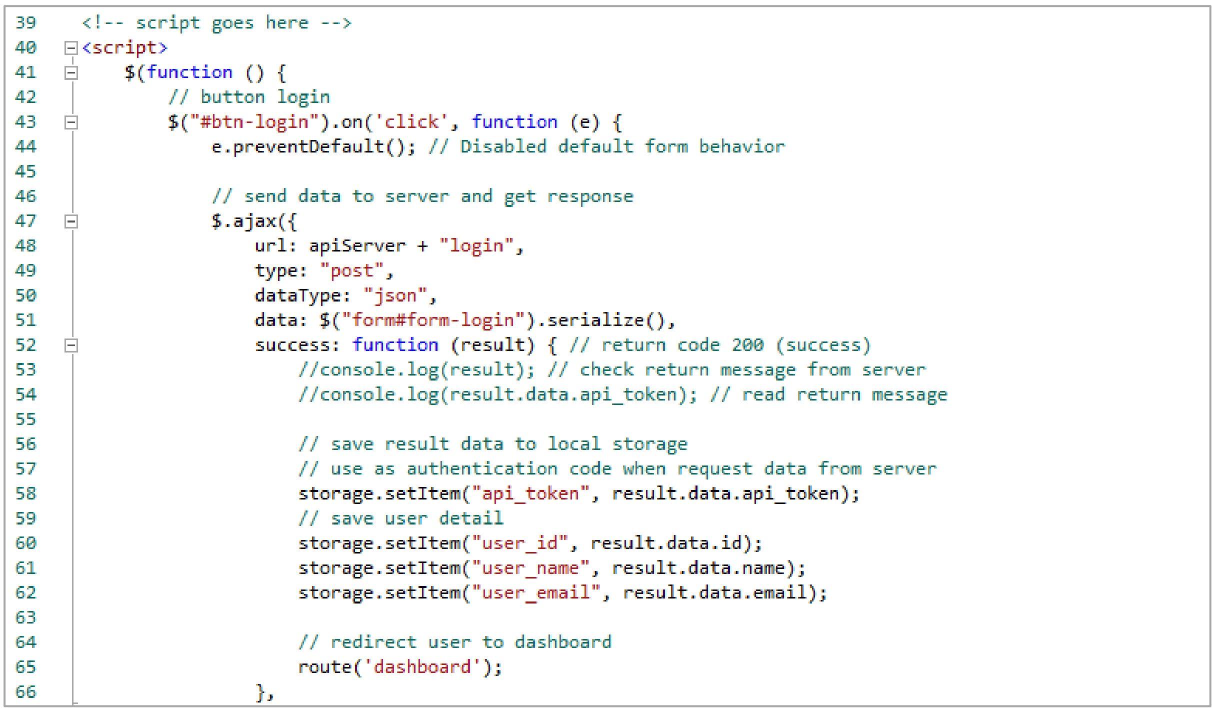
Insert following code in ***login.html***:



```html
<div class="row">
    <div class="col-sm-12 mt-5">
        <h1 align="center"><img src="images/icon-512.png" style="width: 150px;" />
        <br />
        <span class="app_name">My Application</span></h1>
    </div>
</div>

<div class="row">
    <div class="col-sm-12 mt-5">
        <form id="form-login">
            <div class="form-group">
                <label>Email:</label>
                <input type="text" class="form-control" name="email" id="email">
            </div>

            <div class="form-group">
                <label>Password:</label>
                <input type="password" class="form-control" name="password" id="password">
            </div>

            <div class="form-group">
                <button class="form-control btn btn-success" id="btn-login">Login</button>
            </div>
        </form>
    </div>
</div>
```

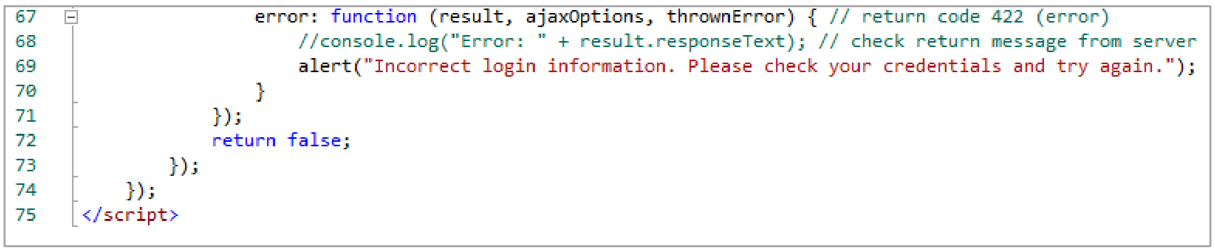
Save and run your project. This code will render login form which have Email and Password field. As this only static page. We will continue our work with some JavaScript & jQuery to make our login page work as required.

Add this code below the finish line:



```html
<!-- script goes here -->
<script>
    $(function () {
        // button login
        $("#btn-login").on('click', function (e) {
            e.preventDefault(); // Disabled default form behavior

            // send data to server and get response
            $.ajax({
                url: apiServer + "login",
                type: "post",
                dataType: "json",
                data: $("form#form-login").serialize(),
                success: function (result) { // return code 200 (success)
                    //console.log(result); // check return message from server
                    //console.log(result.data.api_token); // read return message

                    // save result data to local storage
                    // use as authentication code when request data from server
                    storage.setItem("api_token", result.data.api_token);
                    // save user detail
                    storage.setItem("user_id", result.data.id);
                    storage.setItem("user_name", result.data.name);
                    storage.setItem("user_email", result.data.email);

                    // redirect user to dashboard
                    route('dashboard');
                },
```

Code continue next page…



```
                error: function (result, ajaxOptions, thrownError) { // return code 422 (error)
                    //console.log("Error: " + result.responseText); // check return message from server
                    alert("Incorrect login information. Please check your credentials and try again.");
                }
            });
            return false;
        });
    });
</script>
```

This JavaScript * jQuery code will enable user send data to server. Let's see how it works.

- **`$("#btn-login").on('click', function (e) {});`** will be triggered if the login button *`<button class="form-control btn btn-success" id="btn-login">Login</button>`* from the html part is clicked.
- **`e.preventDefault();`** will disable default form behavior, as in form tag, any button will be triggered as submit button. So this code will prevent that from happened because we want to send our data using REST API via AJAX functionality.
- `$.ajax({ url: apiServer + "login", type: "post", dataType: "json", data: $("form#form-login").serialize() });`
    - `url`: A string containing the URL to which the request is sent..
    - `type`: An alias for method. Either our method want to POST, GET, PUT or DELETE data.
    - `datatype`: Telling jQuery what kind of response to expect. Expecting JSON, or XML, or HTML, etc.
    - `data`: Data to be sent to the server. It is converted to a query string, if not already a string
    - `$("form#form-login").serialize()` : The .serialize() method creates a text string in standard URL-encoded notation. It can act on a jQuery object that has selected individual form controls, such as <input>, <textarea>, and <select>.
- `storage.setItem("user_name", result.data.name);` The read-only localStorage property allows you to access a Storage object for the Document's origin; the stored data is saved across browser sessions and can get back the data using getItem('keyName';)

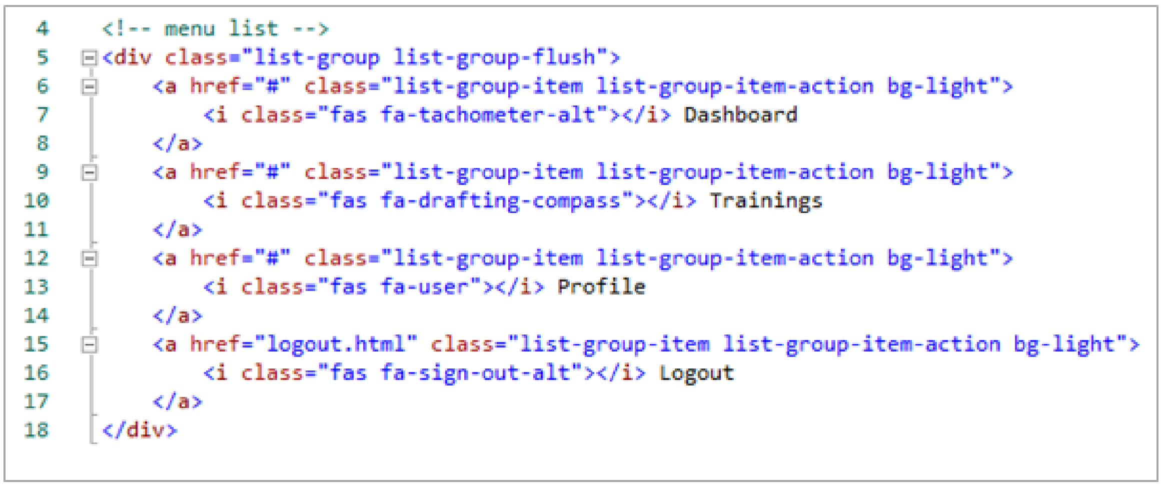
Now refresh your page, and try login using these credentials:

- Email: user@email.me / Password: password

If you can login, then your code works fine. We will add logout functionality to enable user to sign out from the application.

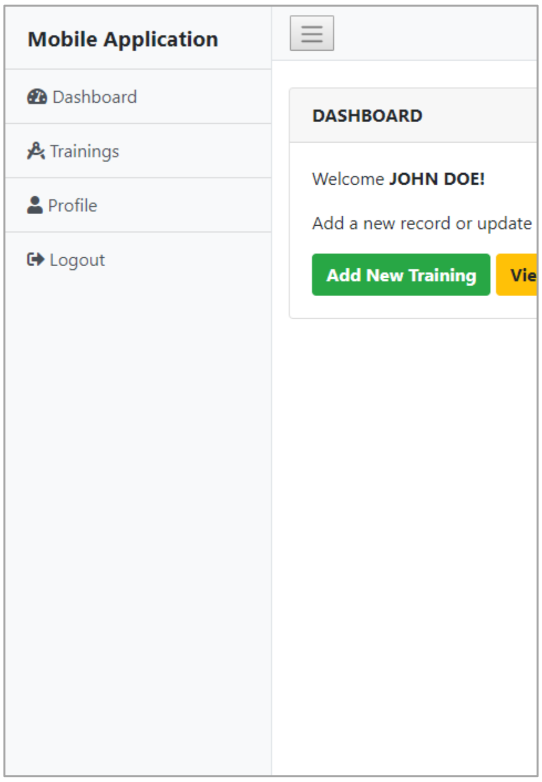
Logout

Open ***sidebar.html*** inside 'view' folder and add this code. This code will create menu item in the sidebar.



```html
<!-- menu list -->
<div class="list-group list-group-flush">
    <a href="#" class="list-group-item list-group-item-action bg-light">
        <i class="fas fa-tachometer-alt"></i> Dashboard
    </a>
    <a href="#" class="list-group-item list-group-item-action bg-light">
        <i class="fas fa-drafting-compass"></i> Trainings
    </a>
    <a href="#" class="list-group-item list-group-item-action bg-light">
        <i class="fas fa-user"></i> Profile
    </a>
    <a href="logout.html" class="list-group-item list-group-item-action bg-light">
        <i class="fas fa-sign-out-alt"></i> Logout
    </a>
</div>
```

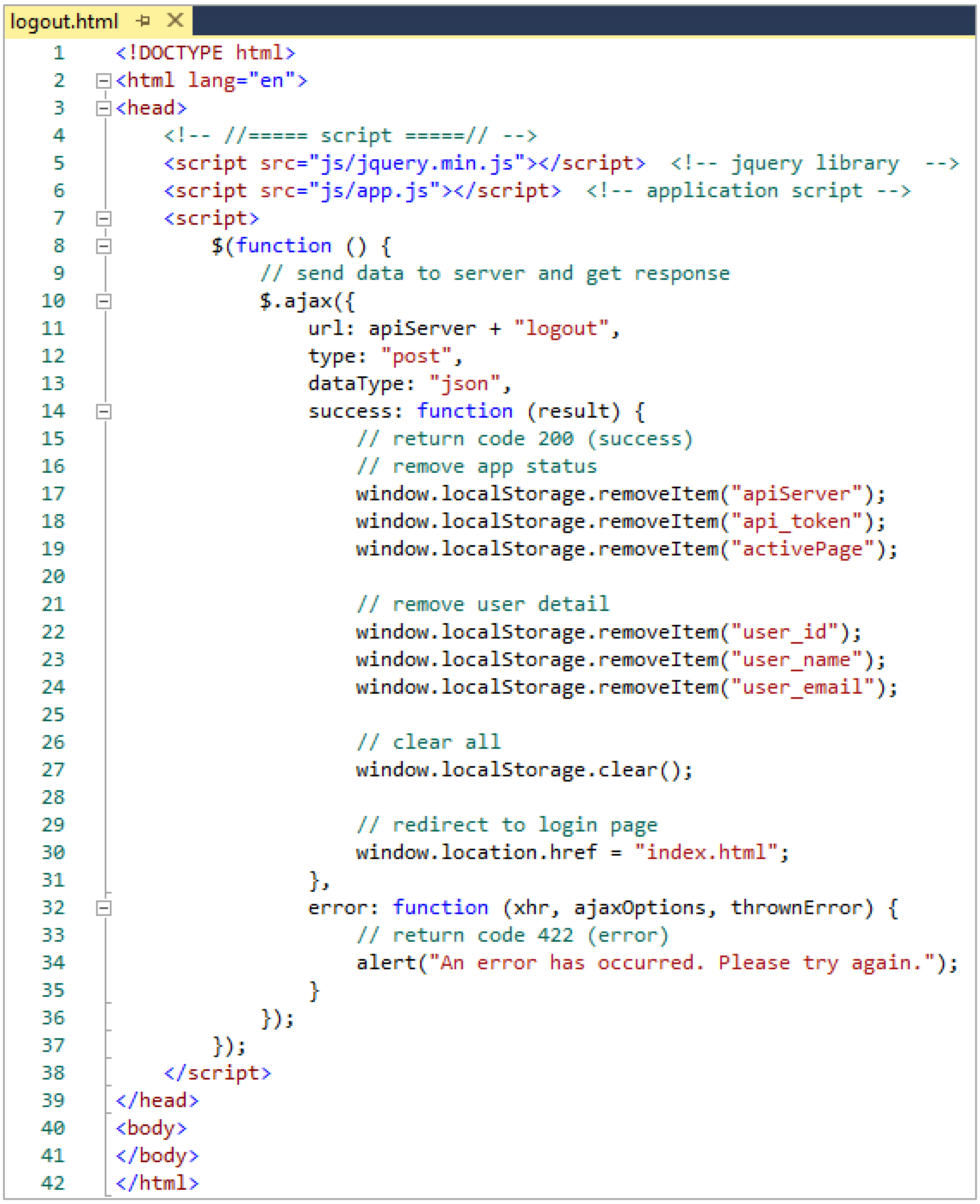
Refresh and see if your menu is now available as below.



Welcome **JOHN DOE!**

Add a new record or update

Next, create ***logout.html*** in 'root (project)' folder, outside from 'view' folder.

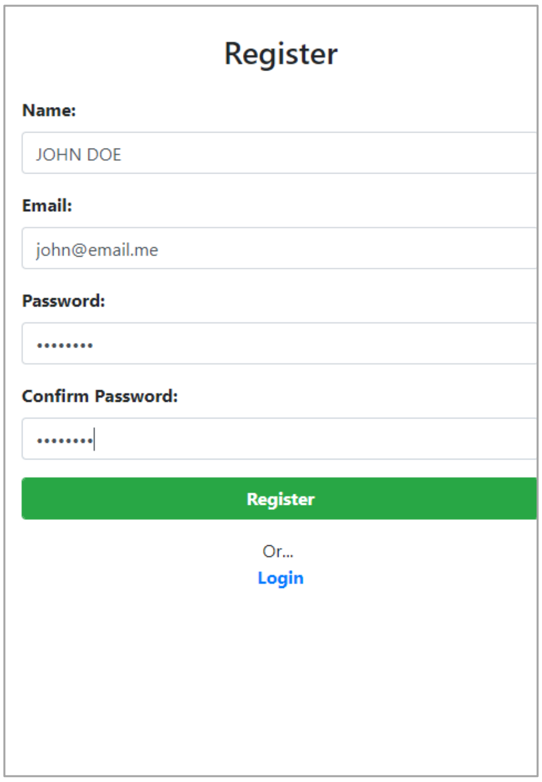

logout.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- //===== script =====// -->
    <script src="js/jquery.min.js"></script>  <!-- jquery library  -->
    <script src="js/app.js"></script>  <!-- application script -->
    <script>
        $(function () {
            // send data to server and get response
            $.ajax({
                url: apiServer + "logout",
                type: "post",
                dataType: "json",
                success: function (result) {
                    // return code 200 (success)
                    // remove app status
                    window.localStorage.removeItem("apiServer");
                    window.localStorage.removeItem("api_token");
                    window.localStorage.removeItem("activePage");

                    // remove user detail
                    window.localStorage.removeItem("user_id");
                    window.localStorage.removeItem("user_name");
                    window.localStorage.removeItem("user_email");

                    // clear all
                    window.localStorage.clear();

                    // redirect to login page
                    window.location.href = "index.html";
                },
                error: function (xhr, ajaxOptions, thrownError) {
                    // return code 422 (error)
                    alert("An error has occurred. Please try again.");
                }
            });
        });
    </script>
</head>
<body>
</body>
</html>
```

The AJAX code will request user logout from the server, as the server will keep the login token. So when user logout, the token will be remove to prevent unauthorized user login using our account.

If result success, application will clear all the localStorage item and redirect to login page.

Tutorial 04 – User Registration



This chapter will cover user registration part. Open ***login.html*** file, and in the middle of the code after last HTML tag, add this item.

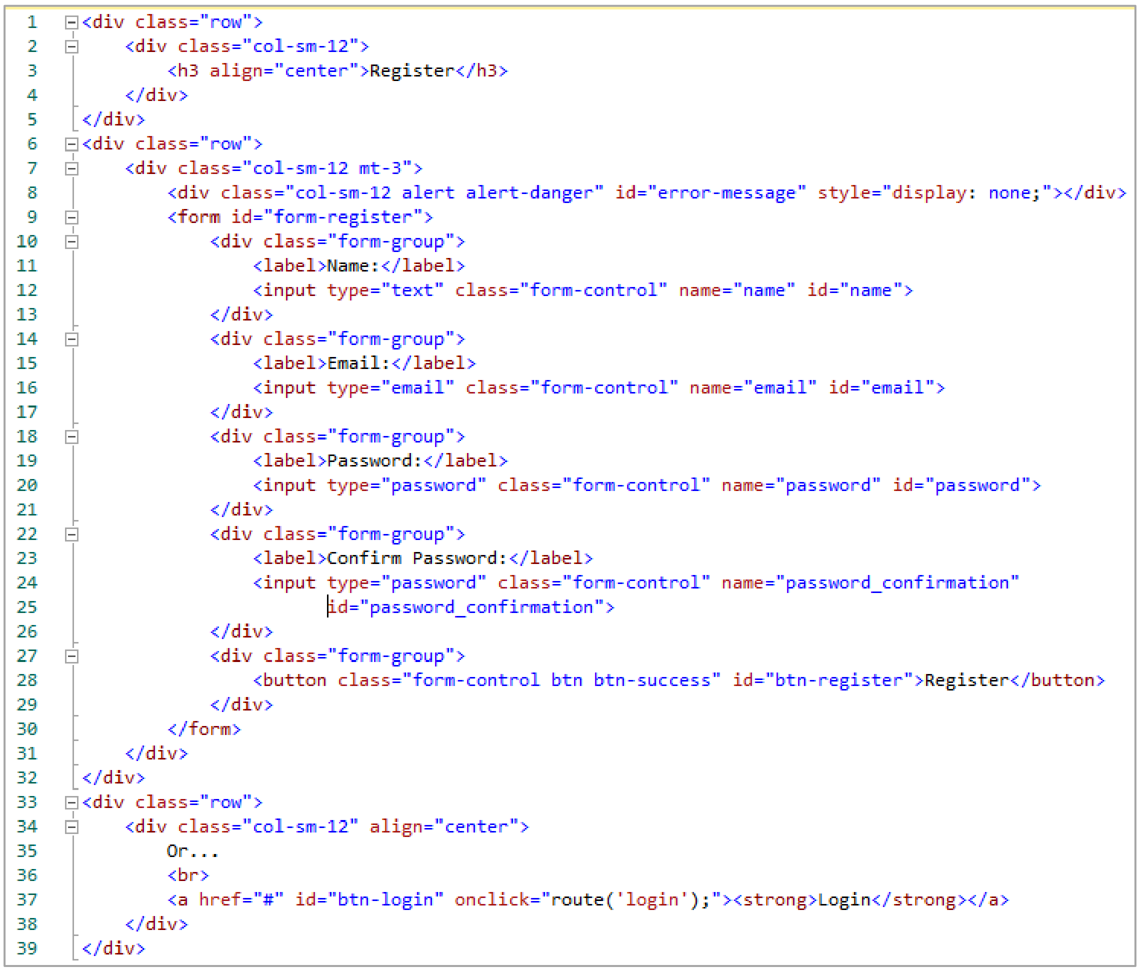
```
<div class="row">
    <div class="col-sm-12" align="center">
        Or...
        <br>
        <a href="#" id="btn-register" onclick="route('register');">
            <strong>Register An Account</strong>
        </a>
    </div>
</div>
```

This code will redirect user to registration page. Please take note that this link is triggered by `onclick="route('register');"` code, not by `href="#"`.

To understand on how the route function, it will search HTML file name that same as the string provided in ‘view’ folder. Eg; `route('register');` will search register.html in the ‘view’ folder.

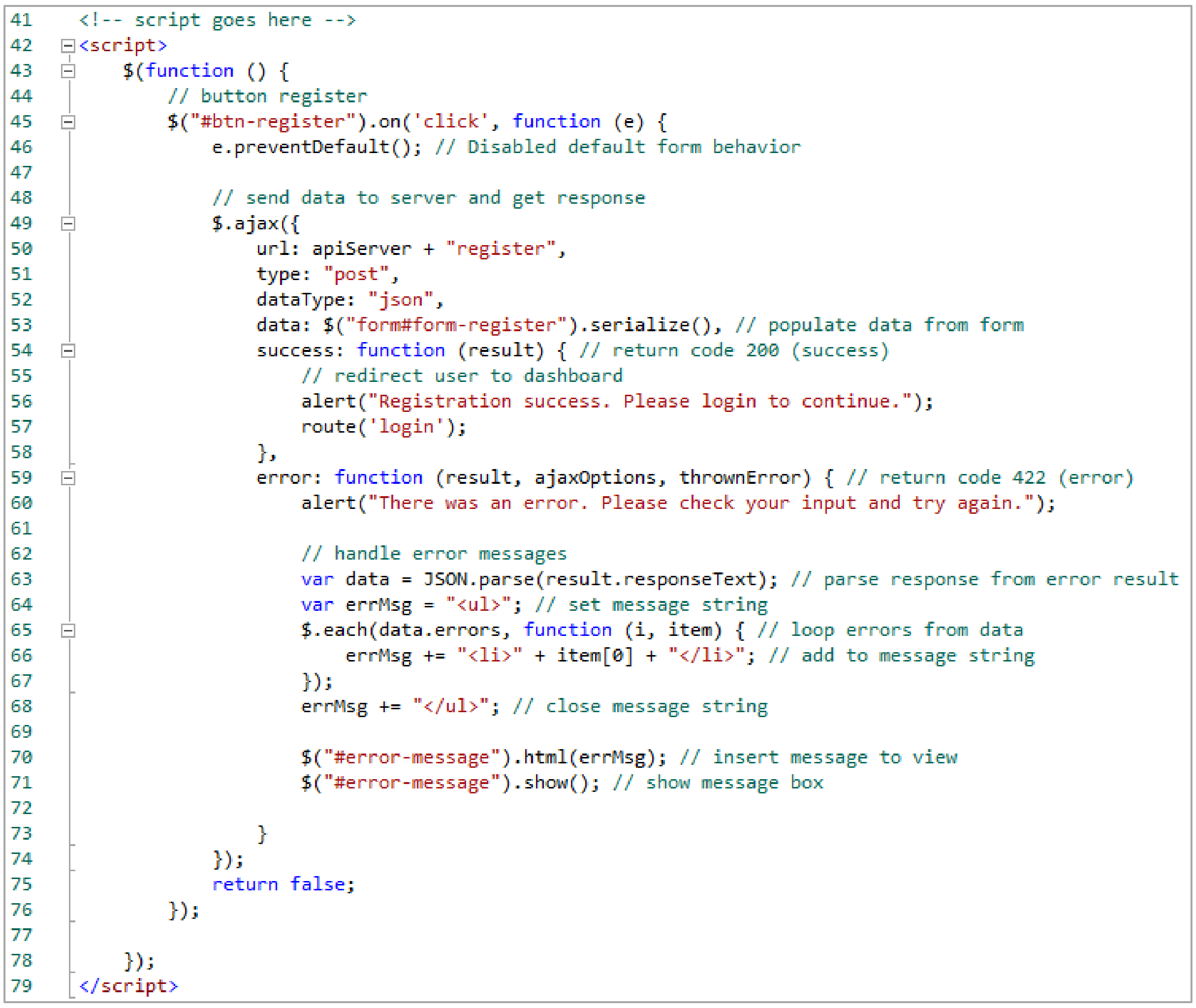
Register

Unfortunately, we still not create the registration file. Lest continue by create new file in ‘view’ folder, name ***register.html*** .



```html
<div class="row">
    <div class="col-sm-12">
        <h3 align="center">Register</h3>
    </div>
</div>
<div class="row">
    <div class="col-sm-12 mt-3">
        <div class="col-sm-12 alert alert-danger" id="error-message" style="display: none;"></div>
        <form id="form-register">
            <div class="form-group">
                <label>Name:</label>
                <input type="text" class="form-control" name="name" id="name">
            </div>
            <div class="form-group">
                <label>Email:</label>
                <input type="email" class="form-control" name="email" id="email">
            </div>
            <div class="form-group">
                <label>Password:</label>
                <input type="password" class="form-control" name="password" id="password">
            </div>
            <div class="form-group">
                <label>Confirm Password:</label>
                <input type="password" class="form-control" name="password_confirmation"
                       id="password_confirmation">
            </div>
            <div class="form-group">
                <button class="form-control btn btn-success" id="btn-register">Register</button>
            </div>
        </form>
    </div>
</div>
<div class="row">
    <div class="col-sm-12" align="center">
        Or...
        <br>
        <a href="#" id="btn-login" onclick="route('login');"><strong>Login</strong></a>
    </div>
</div>
```

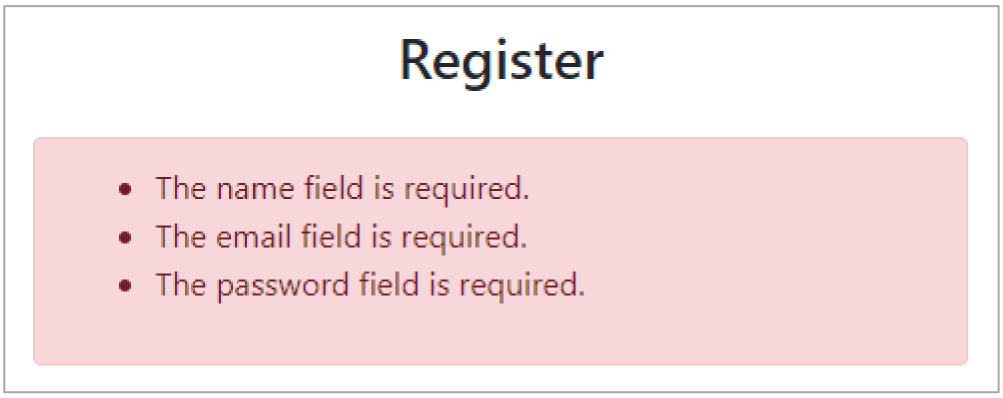
Registration form.



```
41   <!-- script goes here -->
42   <script>
43       $(function () {
44           // button register
45           $("#btn-register").on('click', function (e) {
46               e.preventDefault(); // Disabled default form behavior
47
48               // send data to server and get response
49               $.ajax({
50                   url: apiServer + "register",
51                   type: "post",
52                   dataType: "json",
53                   data: $("form#form-register").serialize(), // populate data from form
54                   success: function (result) { // return code 200 (success)
55                       // redirect user to dashboard
56                       alert("Registration success. Please login to continue.");
57                       route('login');
58                   },
59                   error: function (result, ajaxOptions, thrownError) { // return code 422 (error)
60                       alert("There was an error. Please check your input and try again.");
61
62                       // handle error messages
63                       var data = JSON.parse(result.responseText); // parse response from error result
64                       var errMsg = "<ul>"; // set message string
65                       $.each(data.errors, function (i, item) { // loop errors from data
66                           errMsg += "<li>" + item[0] + "</li>"; // add to message string
67                       });
68                       errMsg += "</ul>"; // close message string
69
70                       $("#error-message").html(errMsg); // insert message to view
71                       $("#error-message").show(); // show message box
72
73                   }
74               });
75               return false;
76           });
77
78       });
79   </script>
```

jQuery functionality.

Refresh and try register. If registration success, it will redirect to login page. But if error occurred either error input form or server-side error, the error message will be handled by the line 62 until 71 as below;

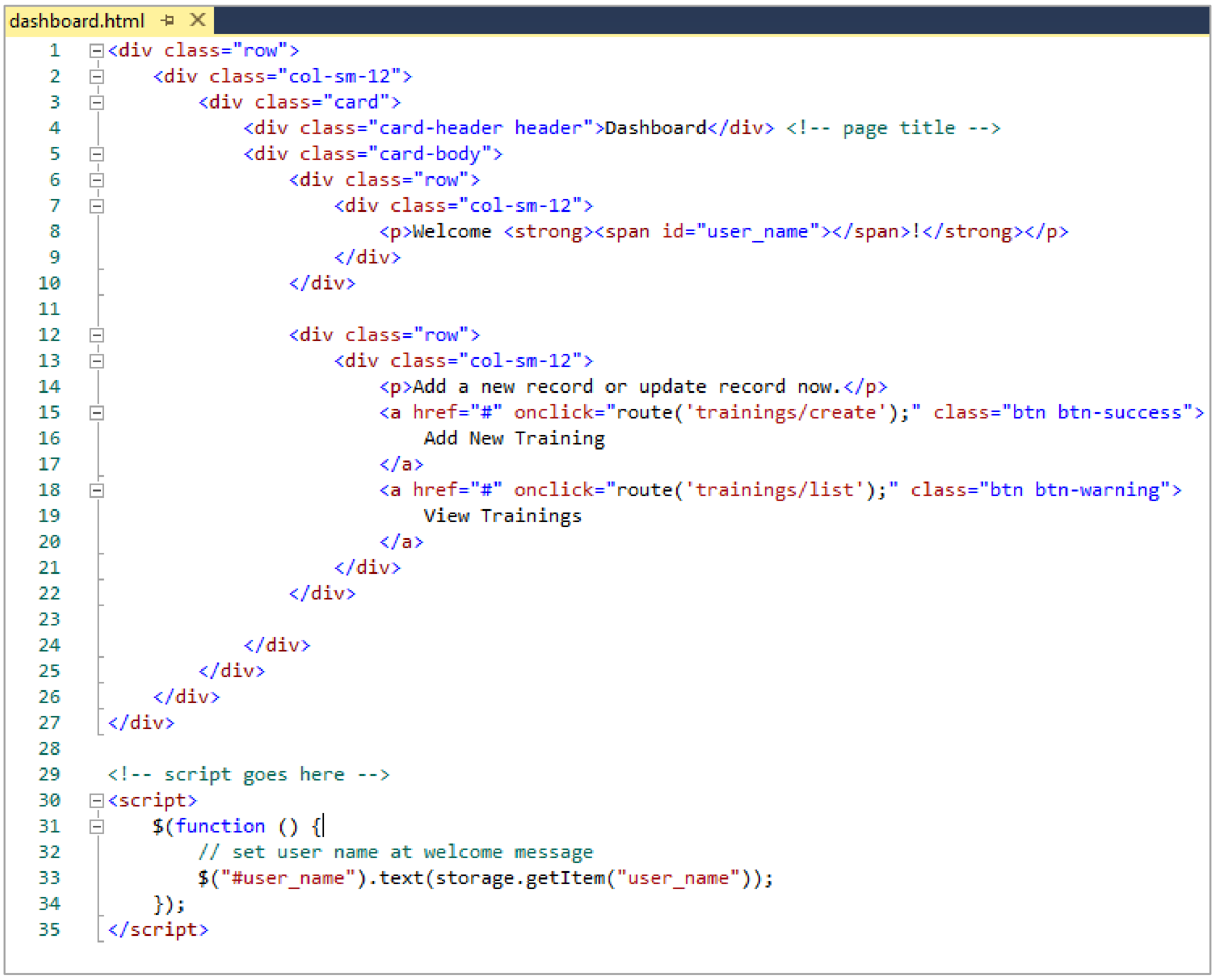


Register

- The name field is required.
- The email field is required.
- The password field is required.

Dashboard

Next, we will set up welcome page after user login. Create new file named ***dashboard.html*** inside ‘view’ folder and enter this code;

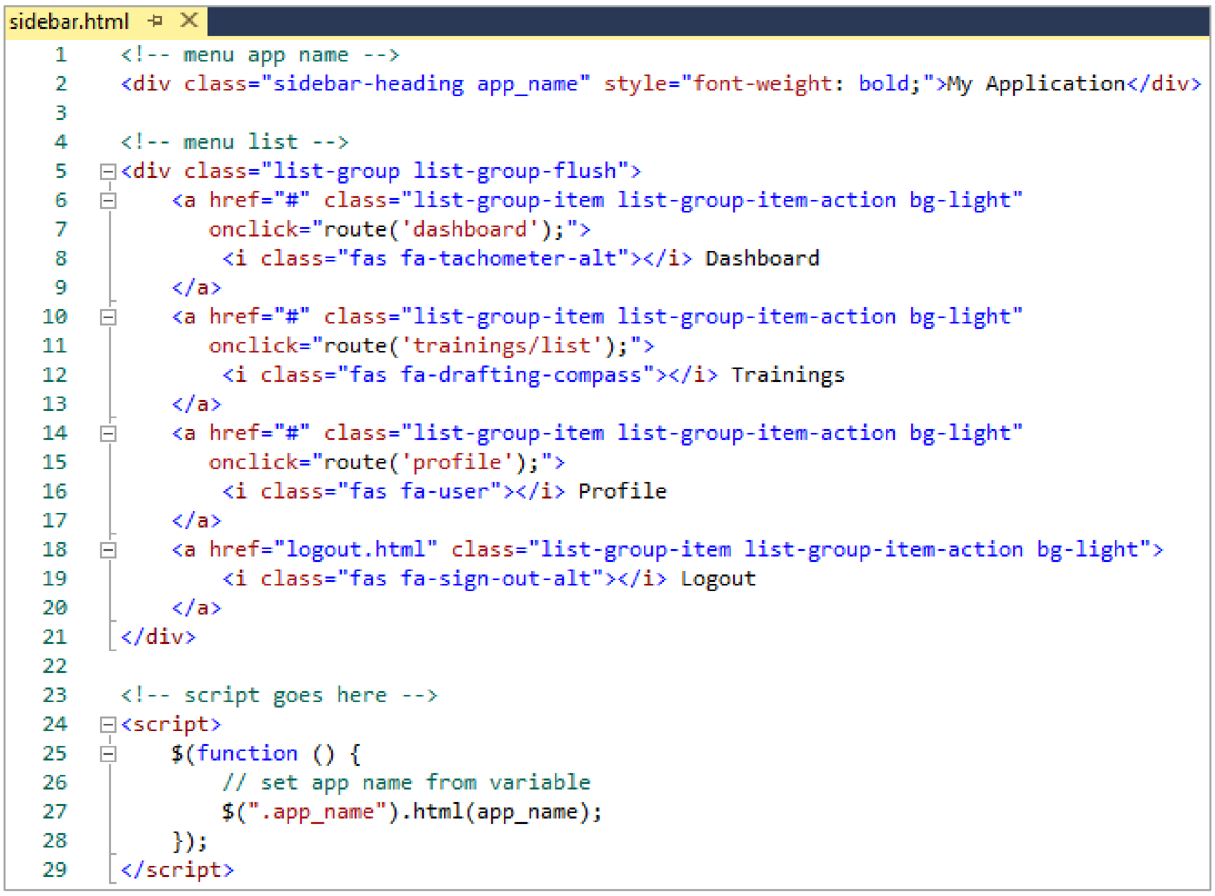


dashboard.html

```
<div class="row">
    <div class="col-sm-12">
        <div class="card">
            <div class="card-header header">Dashboard</div> <!-- page title -->
            <div class="card-body">
                <div class="row">
                    <div class="col-sm-12">
                        <p>Welcome <strong><span id="user_name"></span>!</strong></p>
                    </div>
                </div>

                <div class="row">
                    <div class="col-sm-12">
                        <p>Add a new record or update record now.</p>
                        <a href="#" onclick="route('trainings/create');" class="btn btn-success">
                            Add New Training
                        </a>
                        <a href="#" onclick="route('trainings/list');" class="btn btn-warning">
                            View Trainings
                        </a>
                    </div>
                </div>

            </div>
        </div>
    </div>
</div>

<!-- script goes here -->
<script>
    $(function () {
        // set user name at welcome message
        $("#user_name").text(storage.getItem("user_name"));
    });
</script>
```

This code will have simple welcome message. The JavaScript code write user name that we store in login process to the screen where `<span id="user_name"></span>` is set.

Now, open ***sidebar.html*** again, update the file with following code, where '*onclick*' is added. And JavaScript for print application name in sidebar menu.

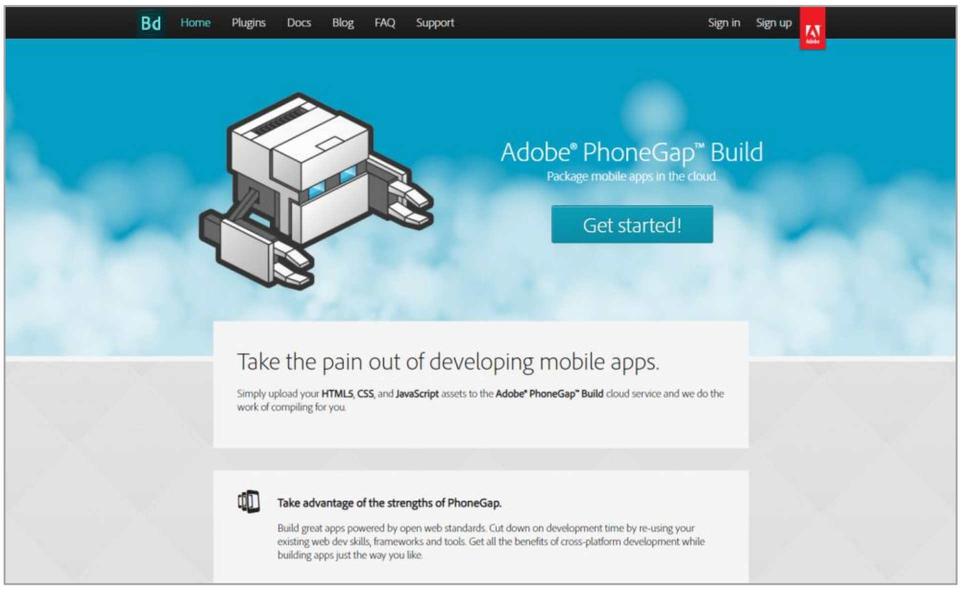

sidebar.html

```
<!-- menu app name -->
<div class="sidebar-heading app_name" style="font-weight: bold;">My Application</div>

<!-- menu list -->
<div class="list-group list-group-flush">
    <a href="#" class="list-group-item list-group-item-action bg-light"
       onclick="route('dashboard');">
        <i class="fas fa-tachometer-alt"></i> Dashboard
    </a>
    <a href="#" class="list-group-item list-group-item-action bg-light"
       onclick="route('trainings/list');">
        <i class="fas fa-drafting-compass"></i> Trainings
    </a>
    <a href="#" class="list-group-item list-group-item-action bg-light"
       onclick="route('profile');">
        <i class="fas fa-user"></i> Profile
    </a>
    <a href="logout.html" class="list-group-item list-group-item-action bg-light">
        <i class="fas fa-sign-out-alt"></i> Logout
    </a>
</div>

<!-- script goes here -->
<script>
    $(function () {
        // set app name from variable
        $(".app_name").html(app_name);
    });
</script>
```

Tutorial 05 – Upload To Adobe PhoneGap Build

In this chapter, we will learn how to upload our application to PhoneGap Build cloud and compile it into mobile installer. Eg; APK file for Android.

1. Open Adobe PhoneGap Build website - https://build.phonegap.com



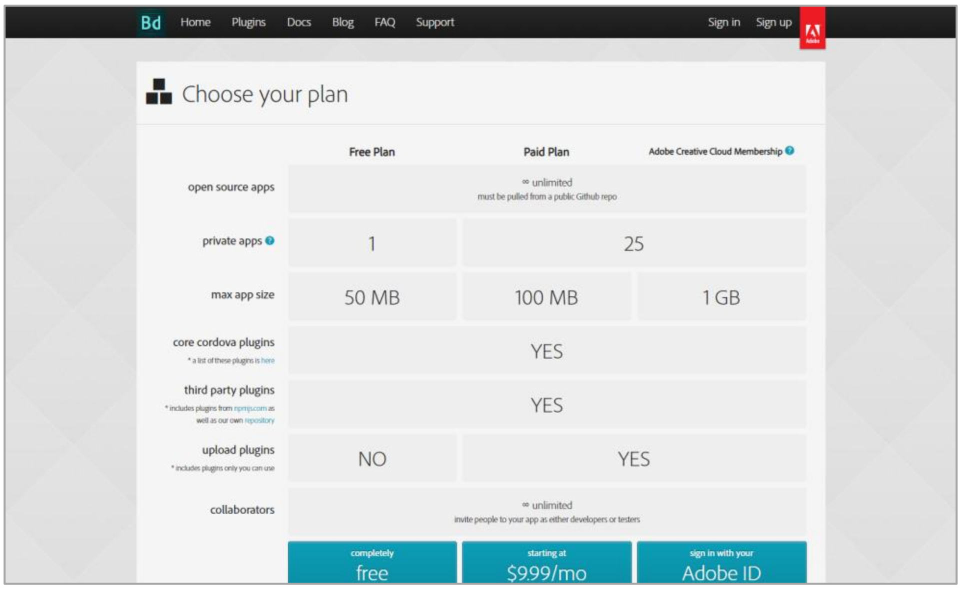
Take the pain out of developing mobile apps.

Simply upload your **HTML5**, **CSS**, and **JavaScript** assets to the **Adobe® PhoneGap™ Build** cloud service and we do the work of compiling for you.

Take advantage of the strengths of PhoneGap.

Build great apps powered by open web standards. Cut down on development time by re-using your existing web dev skills, frameworks and tools. Get all the benefits of cross-platform development while building apps just the way you like.

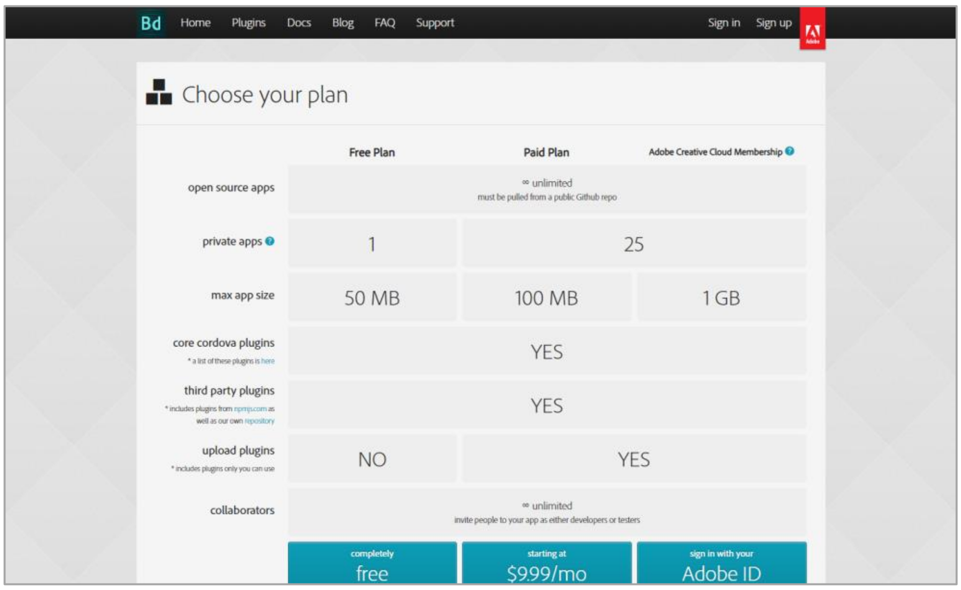
2. Click on **Sign Up** button



Choose your plan

	Free Plan	Paid Plan	Adobe Creative Cloud Membership
open source apps	∞ unlimited must be pulled from a public Github repo		
private apps	1	25	
max app size	50 MB	100 MB	1 GB
core cordova plugins * a list of these plugins is here	YES		
third party plugins * includes plugins from npmjs.com as well as our own repository	YES		
upload plugins * includes plugins only you can use	NO	YES	
collaborators	∞ unlimited invite people to your app as either developers or testers		
	completely free	starting at $9.99/mo	sign in with your Adobe ID

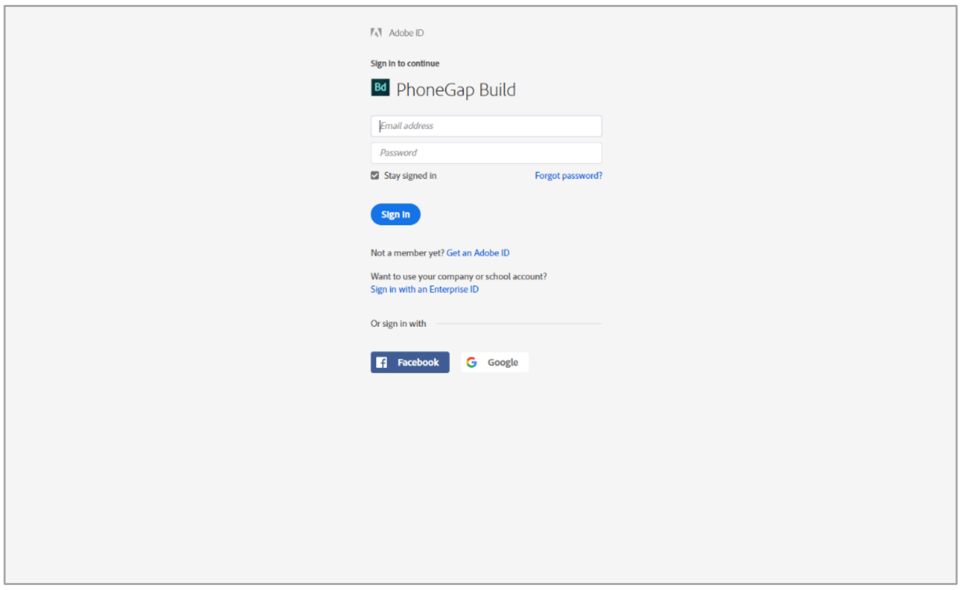
3. Choose **Free Plan**



Choose your plan

	Free Plan	Paid Plan	Adobe Creative Cloud Membership
open source apps	∞ unlimited must be pulled from a public Github repo		
private apps	1	25	
max app size	50 MB	100 MB	1 GB
core cordova plugins * a list of these plugins is here	YES		
third party plugins * includes plugins from npmjs.com as well as our own repository	YES		
upload plugins * includes plugins only you can use	NO	YES	
collaborators	∞ unlimited invite people to your app as either developers or testers		
	completely free	starting at $9.99/mo	sign in with your Adobe ID

4. Register as Adobe member by clicking on **Get an Adobe Id** link.



Sign in to continue

PhoneGap Build

Email address

Password

☑ Stay signed in

Forgot password?

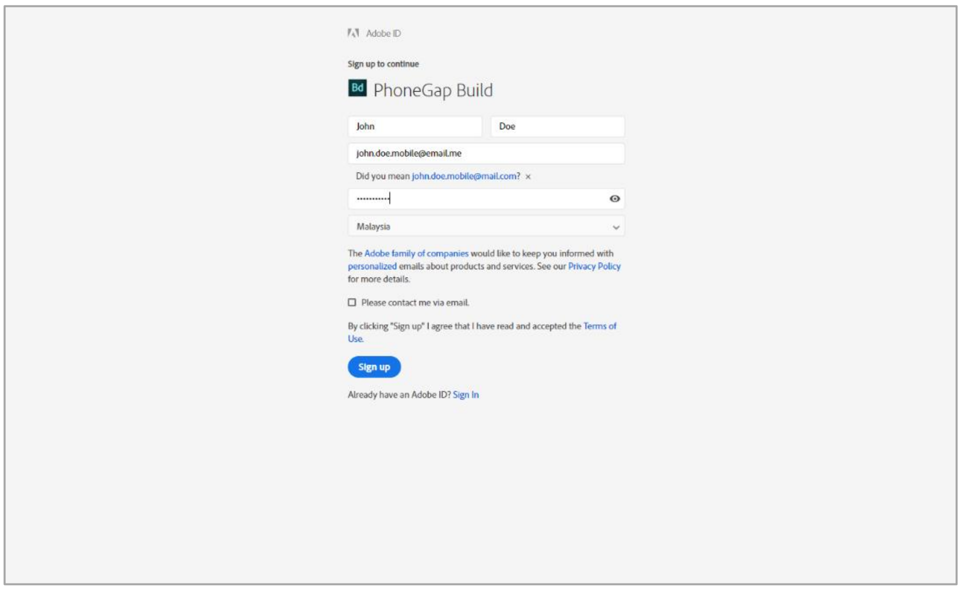
Sign In

Not a member yet? Get an Adobe ID

Want to use your company or school account?
Sign in with an Enterprise ID

Or sign in with

5. Enter account details and click **Sign Up**



John

Doe

john.doe.mobile@email.me

Did you mean john.doe.mobile@mail.com? ×

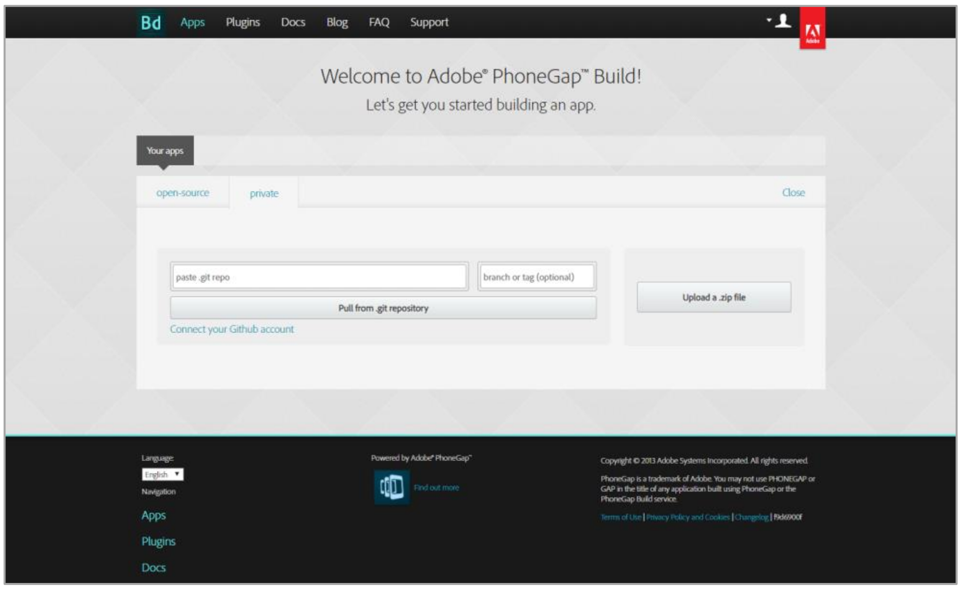
Malaysia

The Adobe family of companies would like to keep you informed with personalized emails about products and services. See our Privacy Policy for more details.

☐ Please contact me via email.

By clicking "Sign up" I agree that I have read and accepted the Terms of Use.

6. You will redirect to Adobe PhoneGap Build dashboard

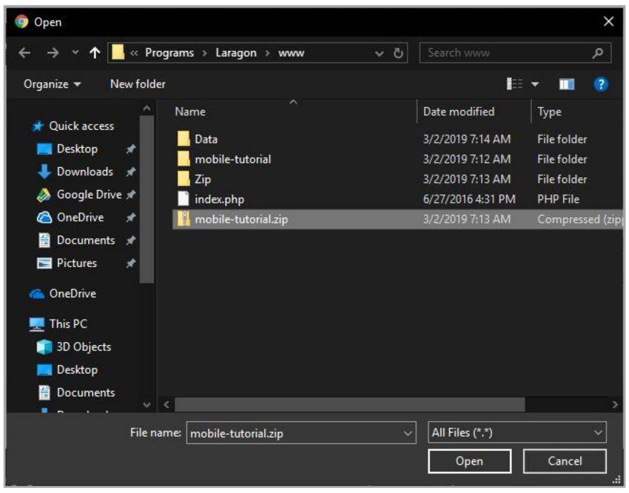


Copyright © 2013 Adobe Systems Incorporated. All rights reserved.

PhoneGap is a trademark of Adobe. You may not use PHONEGAP or GAP in the title of any application built using PhoneGap or the PhoneGap Build service.

Terms of Use | Privacy Policy and Cookies | Changelog | f9d6900f

7. Click **Upload a .zip file.** Select **zipped** project earlier and click **Open**

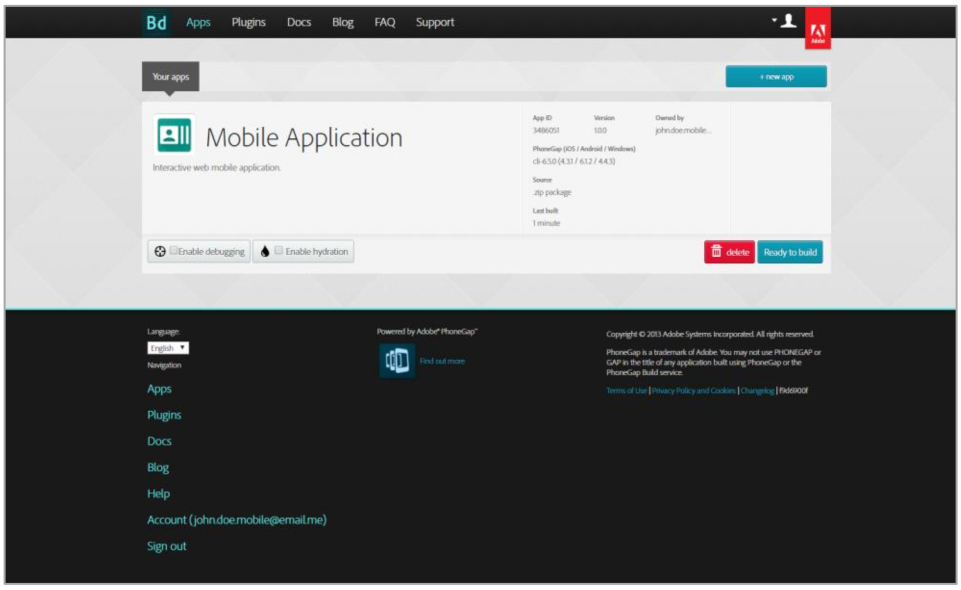


Name	Date modified	Type
Data	3/2/2019 7:14 AM	File folder
mobile-tutorial	3/2/2019 7:12 AM	File folder
Zip	3/2/2019 7:13 AM	File folder
index.php	6/27/2016 4:31 PM	PHP File
mobile-tutorial.zip	3/2/2019 7:13 AM	Compressed (zip

File name: mobile-tutorial.zip

All Files (*.*)

8. Click **Ready to Build**

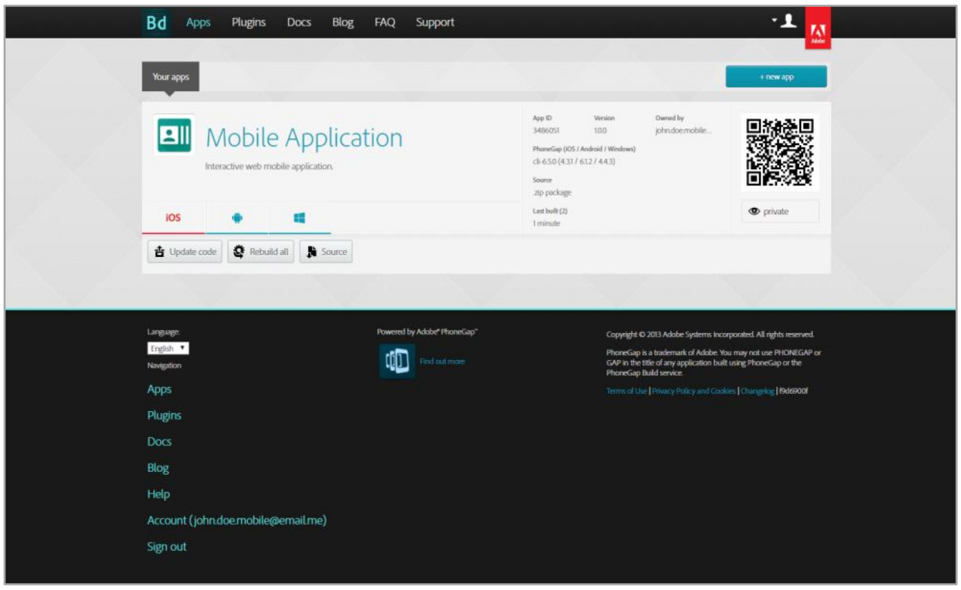


App ID: 3486051
Version: 1.0.0
Owned by: john.doe.mobile...
PhoneGap (iOS / Android / Windows): cli-6.5.0 (4.3.1 / 6.1.2 / 4.4.3)
Source: .zip package
Last built: 1 minute

Copyright © 2013 Adobe Systems Incorporated. All rights reserved.

PhoneGap is a trademark of Adobe. You may not use PHONEGAP or GAP in the title of any application built using PhoneGap or the PhoneGap Build service.

9. If success, you can now download the application installer file by clicking on **Android logo** or scan the **QR Code** from your phone to download directly to phone



App ID: 3486051
Version: 1.0.0
Owned by: john.doe.mobile...
PhoneGap (iOS / Android / Windows): cli-6.5.0 (4.3.1 / 6.1.2 / 4.4.3)
Source: .zip package
Last built (2): 1 minute

Copyright © 2013 Adobe Systems Incorporated. All rights reserved.

PhoneGap is a trademark of Adobe. You may not use PHONEGAP or GAP in the title of any application built using PhoneGap or the PhoneGap Build service.

About the Author

Mahmud Al-Kauthar Mohamad Rabeh
Software Engineer and Mobile Developer
http://www.kauthar.net | https://www.linkedin.com/in/kautharr

Started his carrier as a freelance programmer since 2007, Mahmud is well-experienced in software development. Any tasks of Mobile Application, Web Development and Native Program, he makes them properly to ensure client satisfaction.

Mahmud is have knowledge in various programming language and framework such as .Net, C#, Java, PHP, SQL, Laravel, React, Cordova etc. So far, Mahmud has already developed more than 50 IT projects of web and mobile application.

Mahmud has coordinated and worked closely with his project team in the areas of design, requirement, specification, implementation, coding and testing. From working experience, he has also involved with computer and server troubleshooting and maintenance.